WHAT IS CLAIMED IS:

1. A method of checking the integrity of one or more input vectors to a digital hardware block, comprising the steps of:

identifying a set of known bad input vectors for the digital hardware block; and

training checking circuitry to selectively classify future input vectors to the digital hardware block as either good or not good, using the set of known bad input vectors.

2. The method of Claim 1 further comprising the step of classifying a new input vector of the digital hardware block as not good, using the checking circuitry.

3. The method of Claim 1 wherein said training step trains the checking circuitry to classify as not good both future input vectors which are definitely faulty and future inputs vectors which are potentially faulty.

4. The method of Claim 1 wherein said training step trains the checking circuitry by feeding the set of known bad input vectors to a feedforward linear associative memory neural network.

5. The method of Claim 4 wherein said training step includes the step of creating a weight matrix using a discrete Hopfield network algorithm.

6. The method of Claim 5 wherein said creating step includes the step of calculating the weight matrix W according to the equation

$$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)\,(2b_j^{(m)} - 1)$$

where $a_i^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input vectors in the set of known bad input vectors, i is a row locator representing a particular bad vector, and j is a column locator representing a bit location.

7. The method of Claim 6 wherein said classifying step includes the step of calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$.

8. The method of Claim 7 wherein said classifying step further includes the step of adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i according to the equation

$$\theta i = -\frac{1}{2} \sum_{j=1}^{K} w_{ij}$$

where K is the total number of bits in a vector.

9. The method of Claim 8 wherein said classifying step further includes the step of processing each of the adjusted elements by a respective one of a plurality of non-linear units such that, when a given adjusted element is positive, an output of the corresponding non-linear unit is 1 and, when a given adjusted element is not positive, the output of the corresponding non-linear unit is 0.

10. The method of Claim 2, further comprising the step of executing a software work-around for the new input vector in response to said classifying step.

11. The method of Claim 1, further comprising the step of updating the checking circuitry online.

12. The method of Claim 11 wherein:

the checking circuitry includes a feedforward linear associative memory neural network having a weight matrix W; and

said updating step includes the step of reconfiguring the weight matrix W using one or more additional bad input vectors.

13. A method of providing fault-tolerance in a programmable logic circuit, comprising the steps of:

identifying a hardware block within the programmable logic circuit as being faulty;

providing a software work-around for the faulty hardware block; and

training checking circuitry to selectively classify future input vectors to the faulty hardware block as either good or not good, using a set of known bad input vectors.

14. The method of Claim 13 further comprising the steps of:

handling a new input vector using the faulty hardware block;

classifying the new input vector as not good, using the checking circuitry;

executing a software work-around for the new input vector; and

in response to said classifying step, blocking an output of the faulty hardware block corresponding to the new input vector, and accepting an output of the software work-around corresponding to the new input vector.

15. The method of Claim 13 wherein said training step trains the checking circuitry to classify as not good both future input vectors which are definitely faulty and future inputs vectors which are potentially faulty.

16. The method of Claim 13 wherein said training step trains the checking circuitry by feeding the set of known bad input vectors to a feedforward linear associative memory neural network having a weight matrix W calculated according to the equation

$$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)\,(2b_j^{(m)} - 1)$$

where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input vectors in the set of known bad input vectors, i is a row locator representing a particular bad vector, and j is a column locator representing a bit location.

17. The method of Claim 16 further comprising the step of updating the checking circuitry online, by reconfiguring the weight matrix W using one or more additional bad input vectors.

18. The method of Claim 16 wherein said classifying step includes the further steps of:

calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$;

adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i according to the equation

$$\theta i = -\frac{1}{2} \sum_{j=1}^{K} w_{ij}$$

where K is the total number of bits in a vector; and

processing each of the adjusted elements by a respective one of a plurality of non-linear unit such that, when a given adjusted element is positive, an output of the corresponding non-linear unit is 1 and, when a given adjusted element is not positive, the output of the corresponding non-linear unit is 0.

19. The method of Claim 18 further comprising the step of determining that a vector constructed of the outputs of the non-linear units matches an entry in a memory array storing the set of known bad vectors.

20. A circuit for checking the integrity of one or more input vectors to a digital hardware block, comprising:

a memory array containing a weight matrix having elements which are based on a set of known bad input vectors for the digital hardware block; and

means for selectively classifying future input vectors to the digital hardware block as either good or not good, using the weight matrix.

21. The circuit of Claim 20 further comprising means for creating the weight matrix using a feedforward linear associative memory neural network.

22. The circuit of Claim 20 wherein said classifying means classifies as not good both future input vectors which are definitely faulty and future inputs vectors which are potentially faulty.

23. The circuit of Claim 20 wherein the weight matrix W is created using a discrete Hopfield network algorithm according to the equation

$$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input vectors in the set of known bad input vectors, i is a row locator representing a particular bad vector, and j is a column locator representing a bit location.

24. The circuit of Claim 23 wherein said classifying means includes means for calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$.

25. The circuit of Claim 24 wherein said classifying means further includes means for adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i according to the equation

$$\theta i = -\tfrac{1}{2} \sum_{j=1}^{K} w_{ij}$$

where K is the total number of bits in a vector.

26. The circuit of Claim 25 wherein said classifying means includes a plurality of non-linear units which respectively processes the adjusted elements such that, when a given adjusted element is positive, an output of the corresponding non-

linear unit is 1 and, when a given adjusted element is not positive, the output of the corresponding non-linear unit is 0.

27. The circuit of Claim 20 wherein said classifying means classifies a given input vector as good or not good in less than 60 ns.

28. The circuit of Claim 20 further comprising mean for blocking an output of the digital hardware block corresponding to a new input vector, and accepting an output of a software work-around corresponding to the new input vector, in response to said classifying means classifying the new input vector as not good.

29. The circuit of Claim 20 further comprising means for updating the weight matrix online using one or more additional bad input vectors.

30. The circuit of Claim 27 further comprising:

an SRAM array;

a content-addressable memory for storing the known bad input vectors; and

address management means for using said SRAM to update both said content-addressable memory and said weight matrix.

31. The circuit of Claim 20 wherein said classifying means classifies as not good both future input vectors which are definitely faulty and future inputs vectors which are potentially faulty.

32. The circuit of Claim 20 wherein said classifying means includes a content-addressable memory for storing the known bad input vectors.

33. A programmable logic unit comprising:

a plurality of interconnected hardware blocks, at least one of said hardware blocks being a faulty hardware block;

a feedforward linear associative memory (LAM) neural network checking circuit which classifies input vectors to said faulty hardware block as either good or not good;

a software work-around input; and

a selection circuit connected to said feedforward LAM neural network checking circuit, for (i) blocking an output vector of the faulty hardware block corresponding to a new input vector, (ii) enabling a software work-around for the new input vector, and (iii) accepting an output vector from said software work-around input corresponding to the new input vector, as an output vector of the programmable logic circuit.

34. The programmable logic unit of Claim 33 wherein said selection circuit includes:

a software work-around enable signal output of said feedforward LAM neural network checking circuit; and

a multiplexer having two inputs respectively receiving the output vector from said faulty hardware block and the output vector from said software work-around, and a select line which is connected to said software work-around enable signal output of said feedforward LAM neural network checking circuit.

35. The programmable logic unit of Claim 33 wherein said feedforward LAM neural network checking circuit has a weight matrix whose elements are based on a set of known bad input vectors for said faulty hardware block.

36. The programmable logic unit of Claim 35 wherein said feedforward LAM neural network checking circuit updates the weight matrix online using one or more additional bad input vectors.

37. The programmable logic unit of Claim 36 wherein said feedforward LAM neural network checking circuit includes:

an SRAM array;

a content-addressable memory for storing the known bad input vectors; and

address management means for using said SRAM to update both said content-addressable memory and said weight matrix.

38. The programmable logic unit of Claim 35 wherein the weight matrix W in said feedforward LAM neural network checking circuit is calculated according to the equation

$$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input vectors in the set of known bad input vectors, i is a row locator representing a particular bad vector, and j is a column locator representing a bit location.

39. The programmable logic unit of Claim 38 wherein said feedforward LAM neural network checking circuit includes:

means for calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$;

means for adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i according to the equation

$$\theta i = -\tfrac{1}{2} \sum_{j=1}^{K} w_{ij}$$

where K is the total number of bits in a vector; and

a plurality of non-linear units which process respective elements of the weight matrix to provide an output of 1 when a given adjusted element is positive, and provide an output of 0 when a given adjusted element is not positive.

40. The programmable logic unit of Claim 39 wherein said feedforward LAM neural network checking circuit matches a vector constructed of the outputs of said non-linear units with an entry in a content-addressable memory storing the set of known bad vectors.

41. The programmable logic unit of Claim 33 wherein said feedforward LAM neural network checking circuit classifies input vectors to said faulty hardware block as either good or not good prior to said faulty hardware generating its output vector corresponding to the new input vector.

42. The programmable logic unit of Claim 33 wherein said feedforward LAM neural network checking circuit is physically located in a re-configurable hardware redundancy block.